IPWAVE Working Group   Z. Yan
Internet-Draft   CNNIC
Intended status: Standards Track   J. Lee
Expires: May 14, 2022   Sejong University
                                                              J. Jeong
                                               Sungkyunkwan University
                                                           H. Nakazato
                                                               Y. Park
                                                    Waseda University
                                                    November 10, 2021

Data Aggregation in IPv6-based Vehicular Networks
draft-yan-ipwave-aggregation-04.txt

Abstract

   Considering the large-scale but small-sized information exchange in
   the vehicular information network, this draft document aims at
   outlining the requirements to support the data aggregation in
   vehicular networks based on the concept of Information-centric
   networking (ICN), in order to make the information retrieval and
   dissemination in an efficient way.

Requirements Language

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC 2119.

Status of This Memo

Table of Contents

[1](#). Introduction

   A vehicular information network aims at implementing a myriad of
   applications related to vehicles, traffic information, drivers,
   passengers and pedestrians. Then a flexible data integration and
   segregation architecture in Intelligent Transportation Systems (ITS)
   should be designed to support the exchange of a huge number of
   heterogeneous information objects in an efficient and scalable
   manner.

   The main case for data integration we discuss in this draft is:
   multiple requested information objects originated from different
   sources are shared in some or all hops on the transmission paths.

   This document outlines the general requirements for data integration
   from several key aspects described in the following sections. But
   this draft does not specify the requirements in special communication
   cases, such as Vehicle-to-Everything (V2X), Infrastructure-to-
   Everything (I2X), and Vehicle-to-Infrastructure-to-Vehicle (V2I2V)

communications. The particular requirements under these special
cases will be analyzed in the future.

2. Data naming

Generally, location and data type are potecially critical indexes
for data retrieval in ITS. Also, for configuration, management, and
maintenance, devices may need to be accessed directly by a device-
specific identifier. Therefore, a naming scheme needs to incorporate
location, data type and device information, in order to be scalable
to support trillions of information objects.

o Location-based: A critical organizing factor for vehicular sensing
   data, which is to be widely shared and fused, is the location to
   which it applies.

o Device-based: In some cases, the data produced by a specialized
   vehicle or infrastructure device may be requested.

o Type-based: Another critical element for naming is the type of
   data. Namespaces need also incorporate data type designators,
   such as speed, emission, trajectory and so on.

Then to better support the data aggregation, the name included in the
data request message can be designed as:

/Producer1:Producer2:...ProducerX/ Location1:Location2:...LocationY/
Type1:Type2:...TypeZ/ end/

[The format of the content name used in this document only identifies
the logic of the name structure.]

The parsing logic is: the data objects with Type (1,2,...,Z) created
from Location (1,2,...,Y) by Producer(1,2,...,X) are requested. A
producer identifies the device here.

For example, if a vehicle wants to get the traffic information in
Street-1, Street-2, and Street-3 (without specifying the data
producer/device), a name of the data may be:

//Street-1:Street-2:Street-3/traffic/end/

In most cases, the requester only cares what information it wants,
but does not exactly know the information source. In other words, it
is possible that the requester can not specify the destination
address of the request message. Thus a service discovery scheme,
which may make use of the information in the data name as the index,
can be designed in ITS.

3. **Routing**

   In IP-based vehicular networks, the routing table and routing scope
   should be adaptively designed based on the TCP/IP stack.

   (1) Routing Table

   To support different kinds of ITS communication and different
   aggregation policies, in the routing table of the router in the RSU
   and the edge router in the vehicle, there are at least two types of
   entries to be maintained: geo-location based and IP based routing
   entries. The former one is based on the geographical location
   information of the routers, which is established either through the
   coordinate information exchanged between routers or through
   centralized configuration. On the other hand, the latter one is
   established based on the normal routing protocols in the TCP/IP
   network.

   2) Routing Scope

   As in the IP network, the routing scopes also mainly include
   multicast and unicast for different communication cases. Then
   different routers may be configured for different multicast groups.
   This document mainly considers IPv6 scenario. One router may also
   belong to multiple different multicast groups. Although the data
   aggregation acts like the multicast to converge the communications,
   it is the packet-level optimization and can be applied to both
   unicast and multicast cases.

4. **Aggregation and Segregation**

   Based on the naming labels and the routing information, the router
   (especially a router in an RSU) will decide whether the request
   packet should be split over its multiple outgoing network interfaces
   or not. Specially, the router should determine whether the outgoing
   network interfaces for the multiple data elements the same or not.
   If so, direct forwarding is made based on the matched entry in the
   routing table. Otherwise, the router has to split the original
   request packet into multiple new request packets according to their
   different outgoing network interfaces and send them to different
   next-hop routers according to the newly generated names. Similarly,
   if the data is sent back through the reverse path, they can be
   aggregated.

   As illustrated above, based on the routing table, the router decides
   whether the request message should be split over their related
   outgoing network interfaces or not. However, some conditions (e.g.,
   traffic jam or traffic accident information) should be learned by the

traffic administrator as soon as the vehicular information network
changes quickly and quite frequently. As a result, a timer value
used for the data aggregation should be carefully set. Different
policies for setting the timer value can be used and such policies
need to be indicated by the upper level aggregator (e.g., previous-
hop router) in the request message. Generally, some of the request
messages should be handled on a first-in-first-out basis, for
example, in the emergency case. On the other hand, some of the
request messages can only be processed until all the required
information is collected, for example, in the case where the overall
traffic condition information is required. The upper level
aggregator can set up the timer value to the lower level ones (e.g.,
the next-hop router) in the request message. But the protocol to
support this notification and policy decision is beyond the scope of
this document.

Another key element to support the aggregation and segregation
procedure is a pending table that maintains the original data name
and the newly extracted data names. This table is mainly maintained
by a branching node on the communication path, which conducts the
segregation operation. In this way, the reverse operation (i.e.,
data aggregation) can be executed.

```
 +---+
 | V3|-----\
 +---+     |
           |
       +-----+ //Street-3/traffic/end/
       |RSU3 |------------------\
       +-----+                  |   //Street-3:Street-4/traffic/end/
                          +-----+               +-----+
                          |RSU2 |-----------------|RSU1 |
                          +-----+               +-----+
       +-----+               |                    |
       |RSU4 |-----------------/                   |
       +-----+ //Street-4/traffic/end/             |
           |                                    +---+
 +---+     |                                    |V1 |
 |V4 |-----/                                    +---+
 +---+                     //Street-3:Street-4/traffic/end/
```
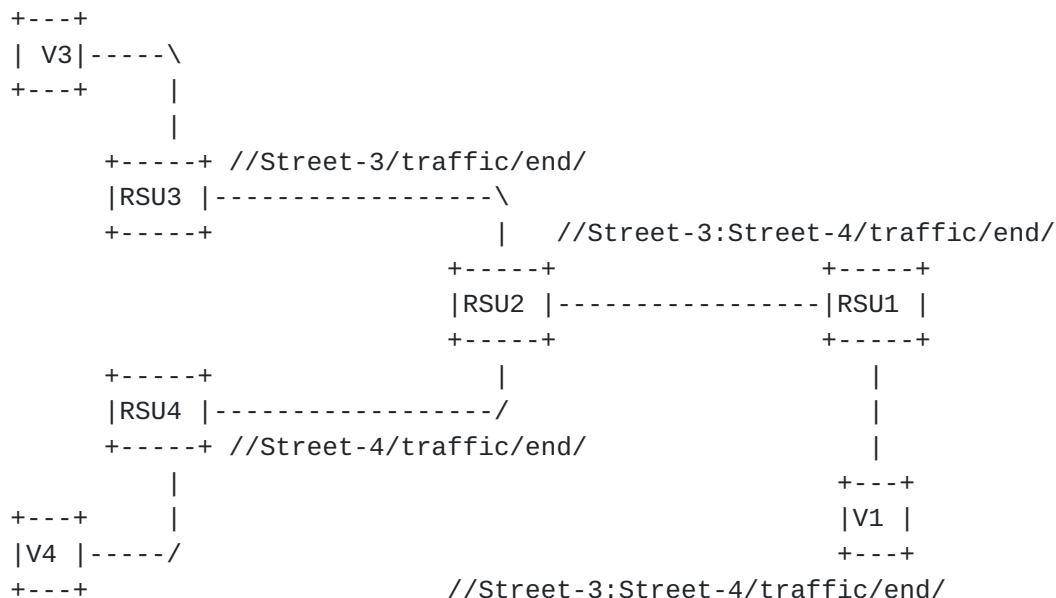
Figure 1: Operation of the Aggregation and Segregation

An example of the aggregation and segregation is shown in Figure 1.
In this figure, Vehicle-1(V1), Vehicle-3(V3), and Vehicle-4(V4)
connect to the Internet through RSU1, RSU3, and RSU4, respectively.
When V1 wants to know the current traffic states of two blocks served

by RSU3 and RSU4 to select a better path between them, it sends out
the data request message with the data name //Street-3:Street-
4/traffic/end/. When RSU1 receives this request message, it directly
sends the message to RSU2 because the next hop to request all the
data in this message comes from RSU2. But when RSU2 receives this
message, it will recognize that the data should be requested from two
different outgoing interfaces toward RSU3 and RSU4, respectively.
Then two new names are generated through the information extraction
from the original name. Specially, the data request for the new name
//Street-3/traffic/end/ is sent to RSU3 and the data request for the
new name //Street-4/traffic/end/ is sent to RSU4.

After the retrieval of the data corresponding to the two data request
messages, the aggregation is conducted through the reverse path based
on the recorded states.

5. Caching

Caching is necessary to reduce unnecessary data transmissions, so it
can improve the scalability in ITS. When the router receives a data
request, it will check its cache firstly. Based on the cache hit
result, the request may be segregated when it is possible.
Generally, two different cache tables should be maintained:

o   Time-sensitive Data Cache: Some data in the ITS is very time-
    sensitive, such as traffic jam condition. Thus, the timer should
    be strictly inherited from the related response message for the
    particular data.

o   Time-insensitive Data Cache: for other time-insensitive data, such
    as the geo-map information, a default timer with a long lifetime
    should be used to serve the following requests efficiently.

6. Other Issues

TBD

7. Security Considerations

TBD

8. Acknowledgements

This work was supported by the Beijing Nova Program of Science and
Technology under grant Z191100001119113.

[9](#). Normative References

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", [BCP 14](#), [RFC 2119](#),
            DOI 10.17487/RFC2119, March 1997,
            <https://www.rfc-editor.org/info/rfc2119>.

Authors' Addresses

   Zhiwei Yan
   CNNIC
   No.4 South 4th Street, Zhongguancun
   Beijing 100190
   China

   EMail: yan@cnnic.cn

   Jong-Hyouk Lee
   Sejong University
   209, Neungdong-ro, Gwangjin-gu
   Seoul 05006
   Republic of Korea

   EMail: jonghyouk@sejong.ac.kr

   Jaehoon Paul Jeong
   Department of Computer Science and Engineering
   Sungkyunkwan University
   2066 Seobu-Ro, Jangan-Gu
   Suwon, Gyeonggi-Do
   Republic of Korea

   EMail: pauljeong@skku.edu

   Hidenori Nakazato
   Waseda University
   1-6-1,Nishi-Waseda,Shinjuku-ku
   Tokyo
   Japan

   EMail: nakazato@waseda.jp

Yong-Jin Park
Waseda University
1-6-1,Nishi-Waseda,Shinjuku-ku
Tokyo
Japan

EMail: yjpark19@gmail.com